

# Diagrama de Componentes

**Régis Patrick Silva Simão**

# Agenda

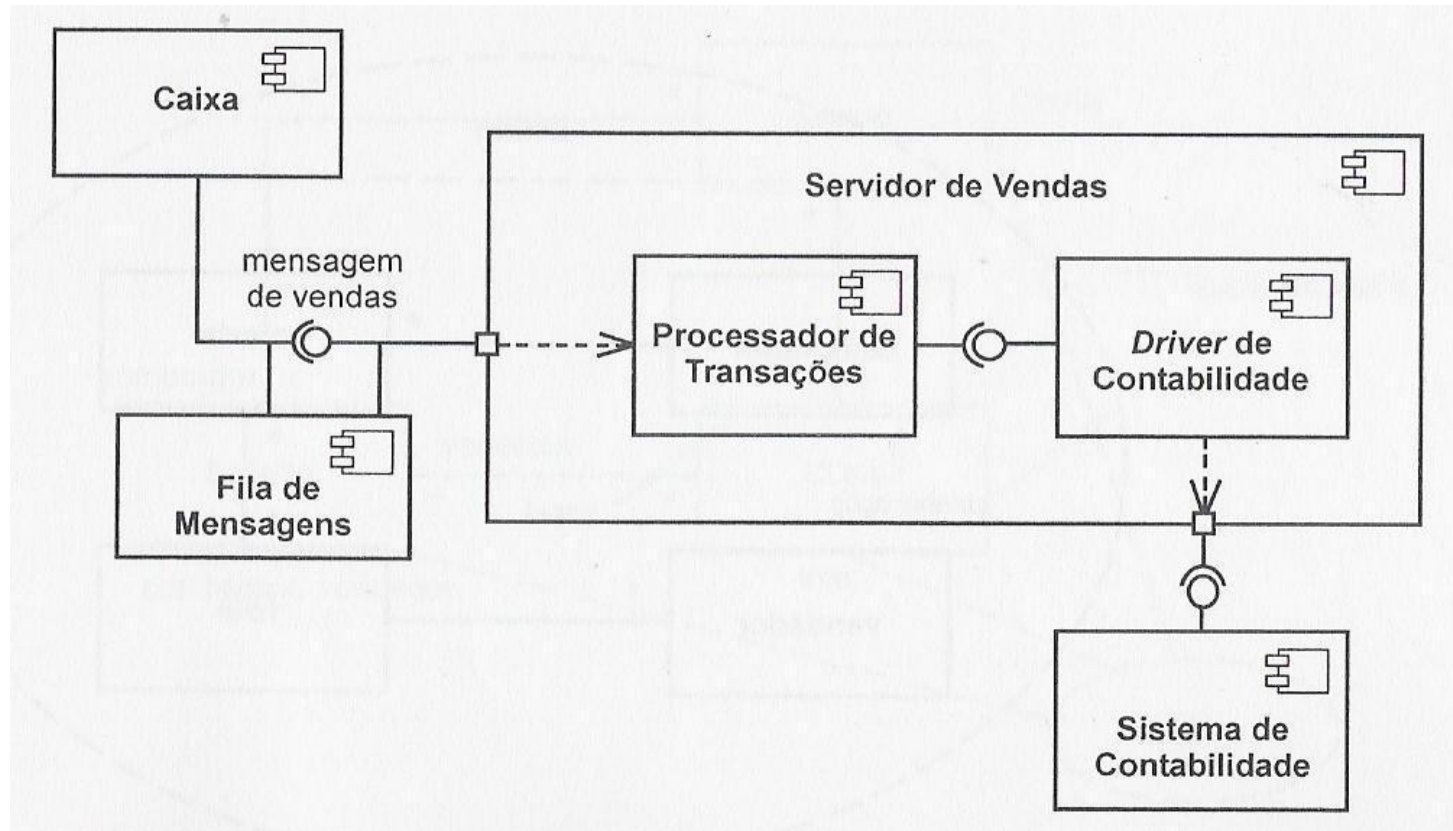
- Introdução
- Componentes
- Interfaces
- Conectores
- Bibliografia

# Introdução

- Na UML 1.x, componentes representam estruturas físicas (arquivos), como executáveis e DLLs. Para esta definição de estruturas físicas, usa-se o termo Artefato.
- Na UML 2.0, componentes representam peças que podem ser adquiridas, atualizadas e substituídas independentemente.
- O Diagrama de Componentes é a representação gráfica dos componentes e de suas relações.
- O diagrama de componentes não apresenta nenhuma notação nova. Todas as notações já foram vistas em outros diagramas.

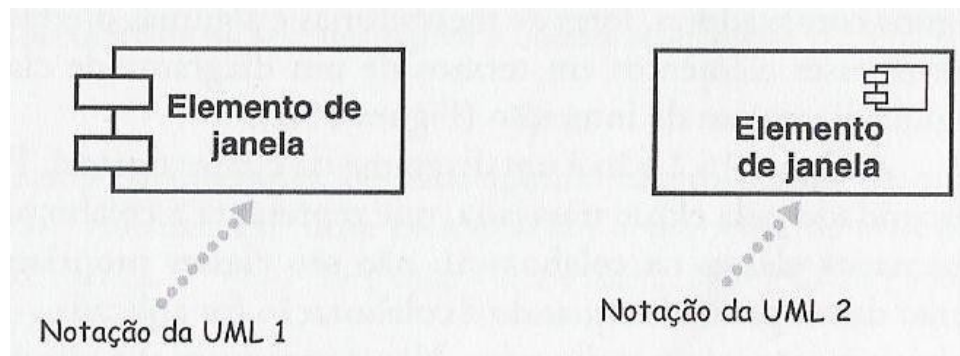
# Introdução

## ➤ Exemplo:



# Componentes

- Um componente é um elemento substituível de um sistema ao qual se adapta e fornece a realização de um conjunto de interfaces.
- Um componente é parte de um sistema. Ele colabora com outros componentes, formando o sistema.
- Um componente é uma implementação empacotada e pronta para uso.



# Componentes

- A representação do componente pode ser feita também por um retângulo com o nome do componente e o estereótipo <<*component*>>.
- Um componente pode possuir interfaces fornecidas e interfaces requeridas. Normalmente, usa-se a notação bola-esquete.
- Um componente é substituível. Pode-se substituir um componente por outro desde que se utilizem as mesmas interfaces.
- Um componente pode ser decomposto usando o Diagrama de Estruturas Compostas. Neste caso, as partes são chamadas de componentes internos e têm todas as características de componente normal.

# Interfaces

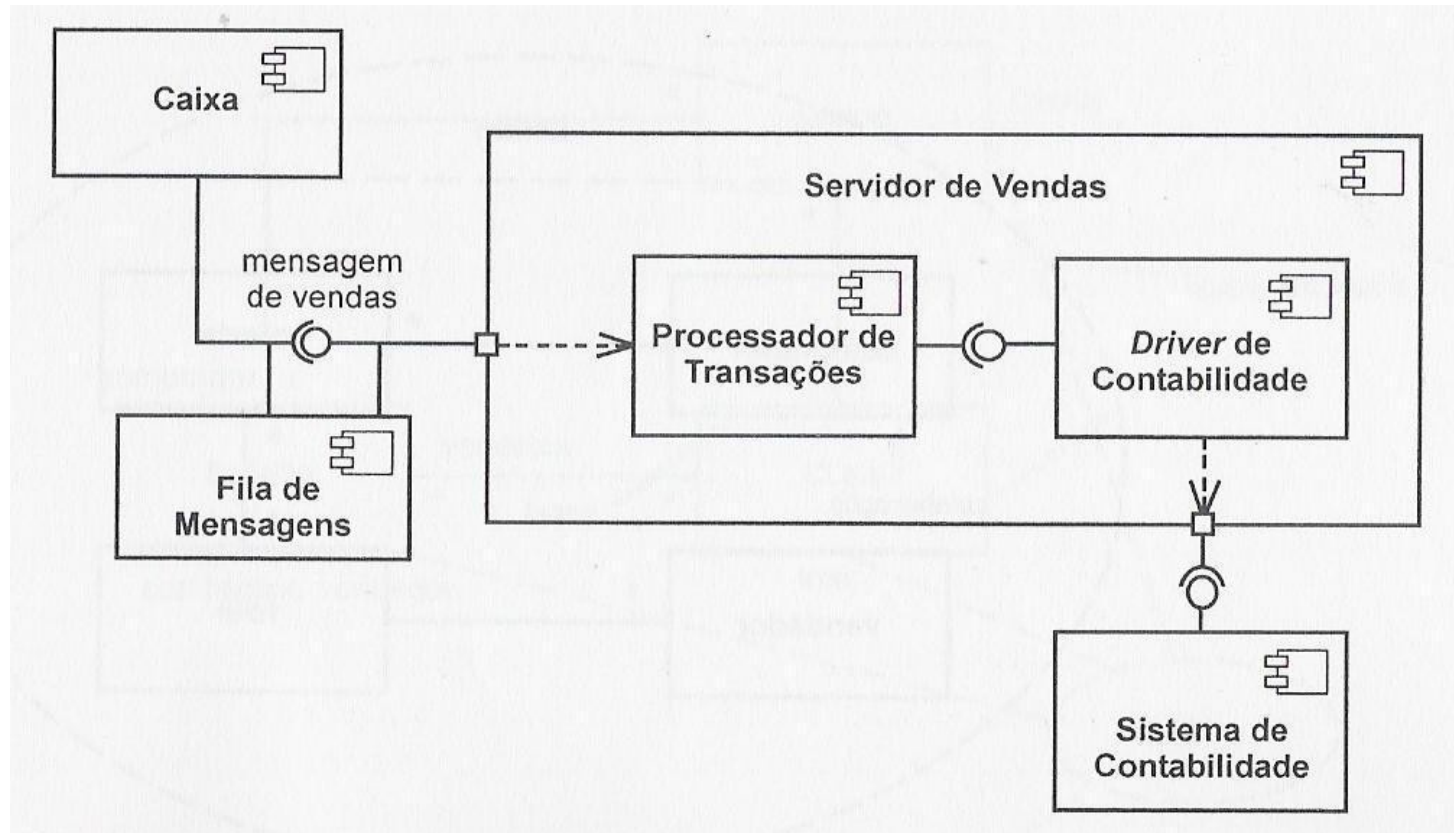
- As interfaces agrupam os comportamentos que são oferecidos ou requeridos pelo componente.
- As interfaces são representadas na notação bola-e-soquete ou listadas internamente ao componente.
- Uma interface oferecida é representada por uma linha saindo do componente para um círculo.
- Uma interface requerida é representada por uma linha saindo do componente para um semi-círculo.

# Conectores

- Um conector (conector de delegação) pode ser usado para mostrar que um componente interno exige uma interface externa ao componente pai, a interface requerida.
  - Neste caso, usa-se a seta de dependência saindo da parte e chegando a interface.
- Um conector (conector de delegação) também é usado para mostrar que componente interno provê uma implementação para uma determinada interface, a interface oferecida.
  - Neste caso, pode ser usada a seta de dependência saindo da interface e chegando componente interno ou pode ser usada a seta de implementação (realização) saindo do componente interno e chegando a interface.



# Conectores



# Conectores

- Um conector pode também ser usado para mostrar a ligação entre os componentes internos. Neste caso, o conector pode ser representado por uma linha entre os dois componentes.
- A notação bola-e-soquete também pode ser usada entre os componentes internos, enfatizando a informação de que parte oferece o serviço. No caso do diagrama de componentes, este estilo é o mais usado.

# Bibliografia

- FOWLER, Martin. UML Essencial. 3a. edição. Bookman, 2005.

**FIM!!!**